EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	51298	((query or querying) same database)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 11:19
L2	5006	database same recovery	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:30
L3	588	"log file" near3 transaction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:35
L4	3537	commit$4 near3 transaction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:35
L5	2009	remov$4 near3 transaction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:35
L6	548	incomplet$4 near3 transaction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:36
L7	1	"lock type" near (detect$4 or "de-asserted" or "de-asserting")	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:37
L8	1666	1 and 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:38
L9	468	8 and 4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:38

EAST Search History

L10	42	3 and 9	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:38
L11	16	5 and 10	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:38
L12	478	6 and "16"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:38
L13	6	6 and 11	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:42
L14	36194	"707"/$.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:43
L15	46106	"709"/$.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:44
L16	78128	14 or 15	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:44
L17	4	13 and 16	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/20 14:44

	Document ID	Kind Codes	Source	Issue Date	Pages
1	US 20050216462 A1		US-PGPUB	20050929	17
2	US 20040030703 A1		US-PGPUB	20040212	44
3	US 7076508 B2		USPAT	20060711	41
4	US 6247023 B1		USPAT	20010612	9

	Title	Abstract
1	System and method for a snapshot query during database recovery	
2	Method, system, and program for merging log entries from multiple recovery log files	
3	Method, system, and program for merging log entries from multiple recovery log files	
4	Method for providing database recovery across multiple nodes	

	Current OR	Current XRef	Retrieval Classif	Inventor
1	707/8			Xiao, Wei
2	707/100			Bourbonnais, Serge et al.
3	707/202	707/101		Bourbonnais; Serge et al.

4	707/202	707/10;
		707/200;
		707/201;
		707/203;
		707/8
				Hsiao; Hui-I et al.

US007076508B2

(12) **United States Patent**
Bourbonnais et al.

(10) **Patent No.: US 7,076,508 B2**
(45) **Date of Patent: Jul. 11, 2006**

(54) **METHOD, SYSTEM, AND PROGRAM FOR MERGING LOG ENTRIES FROM MULTIPLE RECOVERY LOG FILES**

(75) Inventors: **Serge Bourbonnais**, Palo Alto, CA (US); **Elizabeth Belva Hamel**, Morgan Hill, CA (US); **Bruce G. Lindsay**, San Jose, CA (US); **Chengfei Liu**, Rostrevor (AU); **Jens Stankiewitz**, Dorsten (DE); **Tuong Chanh Truong**, San Jose, CA (US)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 405 days.

(21) Appl. No.: **10/217,707**

(22) Filed: **Aug. 12, 2002**

(65) **Prior Publication Data**

US 2004/0030703 A1 Feb. 12, 2004

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/202**; 707/101

(58) **Field of Classification Search** 707/101, 707/102, 202, 201, 8

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,853,843 A *	8/1989	Ecklund	....	707/203
5,440,727 A	8/1995	Bhide et al.	....	711/117
5,491,819 A *	2/1996	Fatzinger et al.	....	707/8
5,499,367 A	3/1996	Bamford et al.	....	707/8
5,628,006 A *	5/1997	Fatzinger et al.	....	707/202
5,675,791 A	10/1997	Bhide et al.	....	707/205
5,878,414 A	3/1999	Hsiao et al.		
5,890,154 A	3/1999	Hsiao et al.	....	707/8
5,946,689 A	8/1999	Yanaka et al.	....	707/10
5,966,706 A *	10/1999	Biliris et al.	....	707/10
5,991,771 A *	11/1999	Falls et al.	....	707/202
6,192,365 B1	2/2001	Draper et al.	....	707/101
6,671,686 B1 *	12/2003	Pardon et al.	....	707/8
6,856,993 B1 *	2/2005	Verma et al.	....	707/101
2005/0138085 A1 *	6/2005	Verma et al.	....	707/202

OTHER PUBLICATIONS

L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System", ACM, vol. 21, No. 7, Jul. 1978.

M. Stonebraker, "The Case for Shared Nothing", University of California Berkley, pp. 1-5.

B. Lindsay, "Data Where You Want It When You Need It", IBM Corporation, DB2 and Business Intelligence Technical Conference, Oct. 2001, Session F-15.

(Continued)

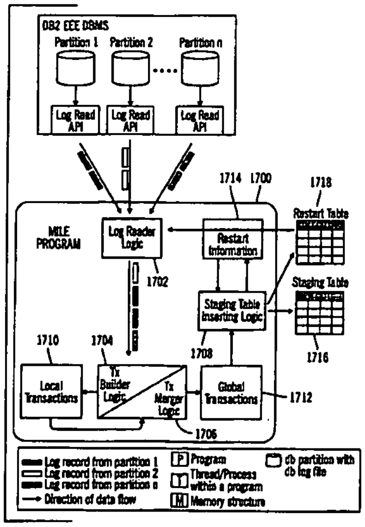
Primary Examiner—Greta Robinson

(74) *Attorney, Agent, or Firm*—Janaki K. Davda; Konrad Raynes & Victor LLP

(57) **ABSTRACT**

Provided are a method, system, and program for merging independent log entries in a multiple node shared nothing DBMS. Initially, log entries from multiple log entries are combined to form a single log entry sequence. Local transactions are generated from the single log entry sequence and stored in a local transactions structure. In particular, log entries with the same local transaction identifier form a local transaction. Then, local transactions having the same global identifier are merged to form global transactions.

27 Claims, 25 Drawing Sheets



US006247023B1

(12) **United States Patent**
Hsiao et al.

(10) **Patent No.: US 6,247,023 B1**
(45) **Date of Patent: Jun. 12, 2001**

(54) **METHOD FOR PROVIDING DATABASE RECOVERY ACROSS MULTIPLE NODES**

(75) Inventors: **Hui-I Hsiao**, Saratoga; **Amy Chang**, San Jose, both of CA (US)

(73) Assignee: **Internationl Business Machines Corp.**, Armonk, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/120,381**

(22) Filed: **Jul. 21, 1998**

(51) **Int. Cl.**^{7} **G06F 12/00**

(52) **U.S. Cl.** **707/202**; 707/8; 707/10; 707/200; 707/201; 707/203

(58) **Field of Search** 707/200, 201, 707/202, 203, 8, 10; 714/4, 19, 20; 709/230; 711/147, 206, 216

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,201,044	*	4/1993	Frey, Jr. et al.	714/20
5,335,343	*	8/1994	Lampson et al.	714/9
5,432,926	*	7/1995	Citron et al.	714/4
5,799,305	*	8/1998	Bortvedt et al.	707/10
6,012,094	*	6/2000	Leymann et al.	709/230

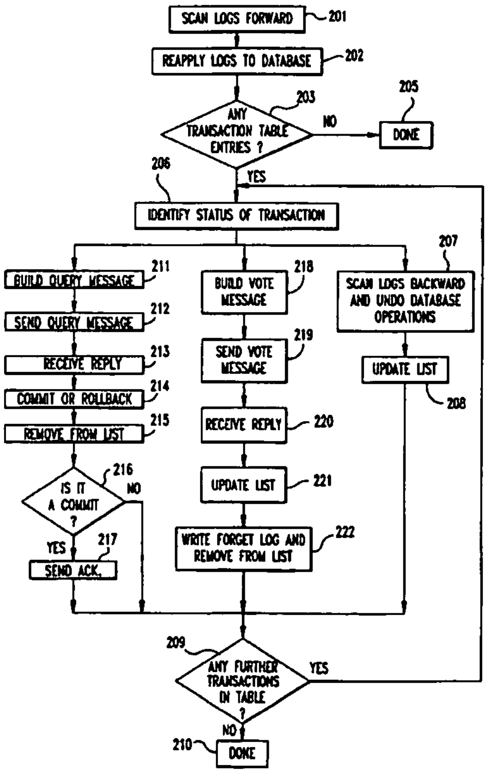
* cited by examiner

Primary Examiner—Thomas Black
Assistant Examiner—Diane D. Mizrahi
(74) *Attorney, Agent, or Firm*—Kevin M. Jordan; David M. Shofi; Anne Vachon Dougherty

(57) **ABSTRACT**

A three-phase database crash recovery mechanism is detailed including a forward phase, a backward phase, and a third, so-called "sideward" phase for recovery of transactions which were interrupted at the time of the crash, using Global Transaction IDs to track the status of the transactions at the coordinating and at the participating nodes. Depending upon the status of the transaction at the time of the crash, either a query message to the coordinating node or a vote message to the participating nodes is generated and processed, thereby allowing most interrupted transactions to be completed. Additional implementations are provided for crash recovery, without duplication of efforts, across multiple nodes in a parallel database environment, for cascaded transactions wherein the database recovery at a local node is triggered by database recovery at a remote node in the parallel system, and for concurrent recovery, wherein database recovery is started concurrently at both transaction coordinator and participant nodes.

22 Claims, 3 Drawing Sheets



US005701480A

United States Patent [19]

Raz

[11] Patent Number: 5,701,480

[45] Date of Patent: *Dec. 23, 1997

[54] DISTRIBUTED MULTI-VERSION COMMITMENT ORDERING PROTOCOLS FOR GUARANTEEING SERIALIZABILITY DURING TRANSACTION PROCESSING

[75] Inventor: Yoav Raz, Newton, Mass.

[73] Assignee: Digital Equipment Corporation, Maynard, Mass.

[*] Notice: The term of this patent shall not extend beyond the expiration date of Pat. No. 5,504,900.

[21] Appl. No.: 47,271

[22] Filed: Apr. 14, 1993

Related U.S. Application Data

[63] Continuation-in-part of Ser. No. 778,254, Oct. 17, 1991, abandoned.

[51] Int. Cl.^{6} G06F 15/00

[52] U.S. Cl. 395/671; 395/182.17

[58] Field of Search 395/700, 650, 395/600, 610, 671, 182.17

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,224,664	9/1980	Trinchieri	364/200
4,249,241	2/1981	Aberle et al.	364/200
4,627,019	12/1986	Ng	364/900
4,881,166	11/1989	Thompson et al.	364/200
5,193,188	3/1993	Franaszek et al.	395/650
5,212,788	5/1993	Lomet et al.	395/600
5,263,156	11/1993	Bowen et al.	395/600
5,369,757	11/1994	Spiro et al.	395/575
5,504,900	4/1996	Raz	395/650

FOREIGN PATENT DOCUMENTS

0457112	11/1991	European Pat. Off.	G06F 9/46

OTHER PUBLICATIONS

H.F. Korth & A. Silberschatz, *Database System Concepts*, McGraw-Hill, Inc., United States, pp. xiv–xix, 490–503 (1991).

Vassos Hadzilacos, "A Knowledge Theoretic Analysis of Atomic Commitment Protocols," Proc. of the Sixth ACM Symposium on Principles of Database Systems, Association for Computing Machinery, New York, NY, Mar. 23–25, 1987, pp. 129–134.

Joseph Y. Halpern, "Using Reasoning about Knowledge to Analyze Distributed Systems," Research Report RJ 5522 (56421) Mar. 3, 1987, Computer Science, IBM Almaden Research Center, San Jose, California, 1987.

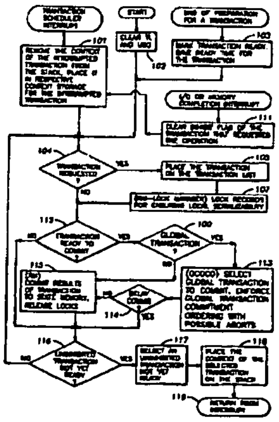
(List continued on next page.)

Primary Examiner—Kevin A. Kriess
Assistant Examiner—Kakali Chaki

[57] **ABSTRACT**

In a multi-version database, copies of prior committed versions (snapshots) are kept for access by the read-only transactions. The read-write transactions are selectively aborted to enforce an order of commitment of read-write transactions that is the same as an order of conflicts among the read-write transactions. In a preferred embodiment, the read-write transactions are serialized by maintaining and referencing a graph of conflicts among read-write transactions, and the read-only transactions are serialized by a timestamp mechanism for selection of the snapshots to be read. Each time that a read-write transaction is committed, the read-write transaction is assigned a unique timestamp that is used to timestamp all resources committed by the read-write transaction. Upon starting, each read-only transaction is also assigned a timestamp. Each read-only transaction reads only the latest committed versions of all resources, that are timestamped earlier than the timestamp of the read-only transaction. In a multiprocessing system, the timestamps are issued to global coordinators and distributed locally with atomic commit messages and global queries. Moreover, read-write transactions may selectively access a hierarchy of uncommitted versions to prepare for various possible commitment orders. The hierarchy defines a path for record access and for cascading aborts. A plurality of mutually-conflicting uncommitted versions may be prepared for each transaction to prepare for all possible commitment orders.

34 Claims, 37 Drawing Sheets


START
101
102
103
104
105
107
108
109
111
112
113
114
115
116
117
118
119
YES
NO
TRANSACTION REQUESTED ?
TRANSACTION READY TO COMMIT ?
GLOBAL TRANSACTION ?
(COCO) SELECT GLOBAL TRANSACTION TO COMMIT, ENFORCE GLOBAL TRANSACTION COMMITMENT ORDERING WITH POSSIBLE ABORTS
PLACE THE TRANSACTION ON THE TRANSACTION LIST
RETURN FROM INTERRUPT

IEEE Xplore®
RELEASE 2.1

Home | Login | Logout | Access Information | Alerts | Sitemap | Help

Welcome United States Patent and Trademark Office

Search Results

BROWSE SEARCH IEEE XPLORE GUIDE SUPPORT

Results for **"(((commit* <near/2> transaction) and snapshot)<in>metadata)"**
Your search matched **1** of **1373978** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

e-mail printer friendly

» Search Options

View Session History

New Search

Modify Search

(((commit* <near/2> transaction) and snapshot)<in>metadata) Search »

□ Check to search only within this results set

Display Format: ● Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

view selected items **Select All** **Deselect All**

□ 1. **A commit protocol for checkpointing transactions**
Pui Ng;
Reliable Distributed Systems, 1988. Proceedings., Seventh Symposium on
10-12 Oct. 1988 Page(s):22 - 31
Digital Object Identifier 10.1109/RELDIS.1988.25777

AbstractPlus | Full Text: PDF(772 KB) **IEEE CNF**
Rights and Permissions

Help Contact Us Privacy & Security IEEE.org

Indexed by Inspec®

Access this document

Full Text: PDF (1012 KB)

Download this citation

Choose Citation & Abstract

Download ASCII Text

» Learn More

Rights and Permissions
» Learn More

Recovery concepts for data sharing systems

Rahm, E.
Dept. of Comput. Sci., Kaiserslautern Univ., Germany;

This paper appears in: Fault-Tolerant Computing, 1991. FTCS-21. Digest of Papers., Twenty-First International Symposium
Publication Date: 25-27 June 1991
On page(s): 368 - 375
Meeting Date: 06/25/1991 - 06/27/1991
Location: Montreal, Que.
INSPEC Accession Number:4031275
Digital Object Identifier: 10.1109/FTCS.1991.146687
Posted online: 2002-08-06 17:44:00.0

Abstract

Crash and media recovery problems that have to be addressed in data sharing systems are addressed. Recovery is complicated by dependencies on other functions such as buffer management and concurrency control. Furthermore, a global **log file** is to be constructed where the modifications of **committed transactions** are reflected in chronological order. Logging and recovery protocols that employ the primary copy approach for concurrency/coherence control are proposed for loosely coupled data sharing systems. A comparison with existing data sharing system shows that the protocols support high performance during normal processing as well as efficient recovery that provides high availability

Index Terms
Inspec
Controlled Indexing
distributed processing fault tolerant computing

Non-controlled Indexing
buffer management chronological order **committed transactions** concurrency control crash recovery data sharing systems **global log file** media recovery primary copy recovery protocols

Author Keywords
Not Available

References

No references available on IEEE Xplore.

Citing Documents

1 Recovery analysis of data sharing systems under deferred dirty page propagation policies, Dan, A.; Yu, P.S.; Jhingran, A.
Parallel and Distributed Systems, IEEE Transactions on
On page(s): 695-711, Volume: 8, Issue: 7, Jul 1997
Abstract | Full Text: PDF (272)

View Search Results

A commit protocol for checkpointing transactions

Pui Ng
Dept. of Comput. Sci., Illinois Univ., Urbana, IL , USA;

This paper appears in: Reliable Distributed Systems, 1988. Proceedings., Seventh Symposium on
Publication Date: 10-12 Oct. 1988
On page(s): 22 - 31
Meeting Date: 10/10/1988 - 10/12/1988
Location: Columbus, OH
INSPEC Accession Number:3328550
Digital Object Identifier: 10.1109/RELDIS.1988.25777
Posted online: 2002-08-06 16:01:10.0

Abstract

A commit protocol is described for checkpointing distributed transactions. Commit protocols are used by distributed transaction management systems to ensure that the multiple nodes participating in a distributed **transaction** will **commit** or abort together. This commit protocol is different from others in that a process executing on behalf of a transaction can be interrupted and restarted at some previous **snapshot** of its state (a checkpoint). The commit protocol guarantees that processes working on behalf of a distributed transaction will be consistent, which implies that the work performed by a restarted process between the time of the checkpoint and the time of the interruption will be undone automatically. The undoing includes any local state changes during that period of time, and any state changes in other processes due to communication with the restarted process in that period. The use of a commit protocol for recovery purposes allows normal execution to be resumed before recovery is completed. Recovery will be carried out in parallel, and the commit protocol guarantees that it is performed eventually. A novel approach of reusing portions of a transaction reduced lost work

Index Terms

Inspec

Controlled Indexing

distributed databases protocols

Non-controlled Indexing

checkpointing transactions commit protocol distributed transactions management systems

Author Keywords

Not Available

References

No references available on IEEE Xplore.

Citing Documents

No citing documents available on IEEE Xplore.

PORTAL

USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

transaction and commit and log file and snapshot and query ar SEARCH

Feedback Report a problem Satisfaction survey

Terms used
transaction and **commit** and **log file** and **snapshot** and **query** and **database** and **lock type** and **de asserted**

Found **28,725** of **182,223**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Transaction management in the R* distributed database management system
C. Mohan, B. Lindsay, R. Obermarck
December 1986 **ACM Transactions on Database Systems (TODS)**, Volume 11 Issue 4
Publisher: ACM Press
Full text available: pdf(1.73 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper deals with the transaction management aspects of the R* distributed database system. It concentrates primarily on the description of the R* commit protocols, Presumed Abort (PA) and Presumed Commit (PC). PA and PC are extensions of the well-known, two-phase (2P) commit protocol. PA is optimized for read-only transactions and a class of multisite update transactions, and PC is optimized for other classes of multisite update transactions. The optimizations resu ...

2 Research sessions: indexing and tuning: Transaction support for indexed summary views
Goetz Graefe, Michael Zwilling
June 2004 **Proceedings of the 2004 ACM SIGMOD international conference on Management of data**
Publisher: ACM Press
Full text available: pdf(168.70 KB) Additional Information: full citation, abstract, references

Materialized views have become a standard technique for performance improvement in decision support databases and for a variety of monitoring purposes. In order to avoid inconsistencies and thus unpredictable query results, materialized views and their indexes should be maintained immediately within user transaction just like indexes on ordinary tables. Unfortunately, the smaller a materialized view is, the higher the concurrency contention between queries and updates as well as among concurrent ...

3 The design of POSTGRES
Michael Stonebraker, Lawrence A. Rowe
June 1986 **ACM SIGMOD Record , Proceedings of the 1986 ACM SIGMOD international conference on Management of data SIGMOD '86**, Volume 15 Issue 2
Publisher: ACM Press
Full text available: pdf(1.91 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents the preliminary design of a new database management system, called POSTGRES, that is the successor to the INGRES relational database system. The main design goals of the new system are to provide better support for complex objects, provide user extendibility for data types, operators and access methods, provide facilities for active databases (i.e., alerters and triggers) and inferencing including forward- ...

4 The Recovery Manager of the System R Database Manager
Jim Gray, Paul McJones, Mike Blasgen, Bruce Lindsay, Raymond Lorie, Tom Price, Franco Putzolu, Irving Traiger
June 1981 **ACM Computing Surveys (CSUR)**, Volume 13 Issue 2
Publisher: ACM Press
Full text available: pdf(1.75 MB) Additional Information: full citation, references, citings, index terms

5 InterViso: dealing with the complexity of federated database access
Marjorie Templeton, Herbert Henley, Edward Maros, Darrel J. Van Buer
April 1995 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 4 Issue 2
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(1.87 MB) Additional Information: full citation, abstract, references, citings

Connectivity products are finally available to provide the "highways" between computers containing data. IBM has provided strong validation of the concept with their "Information Warehouse." DBMS vendors are providing gateways into their products, and SQL is being retrofitted on many older DBMSs to make it easier to access data from standard 4GL products and application development systems. The next step needed for data integration is to provide (1) a common data dictionary with a conceptual sch ...

Keywords: data warehouse, database integration, federated database

6 Heterogeneous distributed database systems for production use
Gomer Thomas, Glenn R. Thompson, Chin-Wan Chung, Edward Barkmeyer, Fred Carter, Marjorie Templeton, Stephen Fox, Berl Hartman
September 1990 **ACM Computing Surveys (CSUR)**, Volume 22 Issue 3
Publisher: ACM Press
Full text available: pdf(2.90 MB) Additional Information: full citation, abstract, references, citings, index terms, review

It is increasingly important for organizations to achieve additional coordination of diverse computerized operations. To do so, it is necessary to have database systems that can operate over a distributed network and can encompass a heterogeneous mix of computers, operating systems, communications links, and local database management systems. This paper outlines approaches to various aspects of heterogeneous distributed data management and describes the characteristics and architectures of ...

7 A new approach to developing and implementing eager database replication protocols
Bettina Kemme, Gustavo Alonso
September 2000 **ACM Transactions on Database Systems (TODS)**, Volume 25 Issue 3
Publisher: ACM Press
Full text available: pdf(449.43 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Database replication is traditionally seen as a way to increase the availability and performance of distributed databases. Although a large number of protocols providing data consistency and fault-tolerance have been proposed, few of these ideas have ever been used in commercial products due to their complexity and performance implications. Instead, current products allow inconsistencies and often resort to centralized approaches which eliminates some of the advantages of replication. As an ...

Keywords: database replication, fault-tolerance, group communication, isolation levels, one-copy-serializability, replica control, total error multicast

8 Final report of the ANSI/X3/SPARC DBS-SG relational database task group

July 1982 **ACM SIGMOD Record**, Volume 12 Issue 4
Publisher: ACM Press
Full text available: pdf(4.69 MB) Additional Information: full citation

9 Distributed transactions for reliable systems
Alfred Z. Spector, Dean Daniels, Daniel Duchamp, Jeffrey L. Eppinger, Randy Pausch
December 1985 **ACM SIGOPS Operating Systems Review , Proceedings of the tenth ACM symposium on Operating systems principles SOSP '85**, Volume 19 Issue 5
Publisher: ACM Press
Full text available: pdf(1.44 MB) Additional Information: full citation, references, citings, index terms

10 Multi-level transaction management for complex objects: implementation, performance, parallelism
Gerhard Weikum, Christof Hasse
October 1993 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 2 Issue 4
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(2.83 MB) Additional Information: full citation, abstract, references, citings

Multi-level transactions are a variant of open-nested transactions in which the subtransactions correspond to operations at different levels of a layered system architecture. They allow the exploitation of semantics of high-level operations to increase concurrency. As a consequence, undoing a transaction requires compensation of completed subtransactions. In addition, multi-level recovery methods must take into consideration that high-level operations are not necessarily atomic if multiple pages ...

Keywords: atomicity, complex objects, inter- and intratransaction parallelism, multi-level transactions, performance, persistence, recovery

11 Research session: architectural issues: C-store: a column-oriented DBMS
Mike Stonebraker, Daniel J. Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Sam Madden, Elizabeth O'Neil, Pat O'Neil, Alex Rasin, Nga Tran, Stan Zdonik
August 2005 **Proceedings of the 31st international conference on Very large data bases VLDB '05**
Publisher: VLDB Endowment
Full text available: pdf(210.85 KB) Additional Information: full citation, abstract, references, index terms

This paper presents the design of a read-optimized relational DBMS that contrasts sharply with most current systems, which are write-optimized. Among the many differences in its design are: storage of data by column rather than by row, careful coding and packing of objects into storage including main memory during query processing, storing an overlapping collection of column-oriented projections, rather than the current fare of tables and indexes, a non-traditional implementation of transactions ...

12 A survey of current object-oriented databases
Mansour Zand, Val Collins, Dale Caviness
February 1995 **ACM SIGMIS Database**, Volume 26 Issue 1
Publisher: ACM Press
Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, citings, index terms

Object-oriented concepts form a good basis for the data models required for next-generation database applications such as CAD/CAE/CASE/CAM systems, knowledge-based systems, multimedia, etc. Many object-oriented databases are available commercially or are being developed by industry or academic research facilities. This paper attempts to compare some of these products using fourteen criteria. The selected criteria are major factors required for the successful design of an object-oriented database ...

Keywords: OOD-BMS survey, object-oriented database, object-oriented terminology

13 Update propagation strategies to improve freshness in lazy master replicated databases
Esther Pacitti, Eric Simon
February 2000 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 8 Issue 3-4
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(151.35 KB) Additional Information: full citation, abstract, index terms

Many distributed database applications need to replicate data to improve data availability and query response time. The two-phase commit protocol guarantees mutual consistency of replicated data but does not provide good performance. Lazy replication has been used as an alternative solution in several types of applications such as on-line financial transactions and telecommunication systems. In this case, mutual consistency is relaxed and the concept of freshness is used to measure the deviation ...

Keywords: Data replication, Distributed databases, Performance evaluation

14 Technical reports
SIGACT News Staff
January 1980 **ACM SIGACT News**, Volume 12 Issue 1
Publisher: ACM Press
Full text available: pdf(5.28 MB) Additional Information: full citation

15 Transactional client-server cache consistency: alternatives and performance
Michael J. Franklin, Michael J. Carey, Miron Livny
September 1997 **ACM Transactions on Database Systems (TODS)**, Volume 22 Issue 3
Publisher: ACM Press
Full text available: pdf(452.41 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Client-server database systems based on a data shipping model can exploit client memory resources by caching copies of data items across transaction boundaries. Caching reduces the need to obtain data from servers or other sites on the network. In order to ensure that such caching does not result in the violation of transaction semantics, a transactional cache consistency maintenance algorithm is required. Many such algorithms have been proposed in the literature and, as all provide the sam ...

16 Garbage collection for a client-server persistent object store
Laurent Amsaleg, Michael J. Franklin, Olivier Gruber
August 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 3
Publisher: ACM Press
Full text available: pdf(267.18 KB) Additional Information: full citation, abstract, references, citings, index terms, review

We describe an efficient server-based algorithm for garbage collecting persistent object stores in a client-server environmnet. The algorithm is incremental and runs concurrently with client transactions. Unlike previous algorithms, it does not hold any transactional locks on data and does non require callbacks to clients. It is fault-tolerant, but performs very little logging. The algorithm has been designed to be integrated into existing systems, and therefore it works with standard i ...

Keywords: client-server system, logging, persistent object-store, recovery

17 Disconnection modes for mobile databases
Joanne Holliday, Divyakant Agrawal, Amr El Abbadi
July 2002 **Wireless Networks,** Volume 8 Issue 4
Publisher: Kluwer Academic Publishers
Full text available: pdf(198.57 KB) Additional Information: full citation, abstract, references, citings, index terms

As mobility permeates into todays computing and communication arena, we envision application infrastructures that will increasingly rely on mobile technologies. Traditional database applications and information service applications will need to integrate mobile entities: people and computers. In this paper, we develop a distributed database framework for mobile environments. A key requirement in such an environment is to support frequent connection and disconnection of database sites. We present ...

Keywords: data consistency, databases, disconnected operation, mobility, replication

18 The LHAM log-structured history data access method
Peter Muth, Patrick O'Neil, Achim Pick, Gerhard Weikum
February 2000 **The VLDB Journal — The International Journal on Very Large Data Bases,** Volume 8 Issue 3-4
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(494.76 KB) Additional Information: full citation, abstract, index terms

Numerous applications such as stock market or medical information systems require that both historical and current data be logically integrated into a temporal database. The underlying access method must support different forms of "time-travel" queries, the migration of old record versions onto inexpensive archive media, and high insertion and update rates. This paper presents an access method for transaction-time temporal data, called the log-structured history data access method (L ...

Keywords: Data warehouses, Index structures, Performance, Storage systems, Temporal databases

19 Multidatabase systems: Overview of multidatabase transaction management
Yuri Breitbart, Hector Garcia-Molina, Avi Silberschatz
November 1992 **Proceedings of the 1992 conference of the Centre for Advanced Studies on Collaborative research - Volume 2**
Publisher: IBM Press
Full text available: pdf(3.06 MB) Additional Information: full citation, abstract, references

A multidatabase system (MDBS) is a facility that allows users access to data located in multiple autonomous database management systems (DBMSs). In such a system, *global transactions* are executed under the control of the MDBS. Independently, *local transactions* are executed under the control of the local DBMSs. Each local DBMS integrated by the MDBS may employ a different transaction management scheme. In addition, each local DBMS has complete control over all transactions (global a ...

20 Overview of multidatabase transaction management
Yuri Breitbart, Hector Garcia-Molina, Avi Silberschatz
October 1992 **The VLDB Journal — The International Journal on Very Large Data Bases,** Volume 1 Issue 2
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(3.23 MB) Additional Information: full citation, abstract, references, citings

A multidatabase system (MDBS) is a facility that allows users access to data located in multiple autonomous database management systems (DBMSs). In such a system, *global transactions* are executed under the control of the MDBS. Independently, *local transactions* are executed under the control of the local DBMSs. Each local DBMS integrated by the MDBS may employ a different transaction management scheme. In addition, each local DBMS has complete control over all transactions (global a ...

Keywords: multidatabase, recovery, reliability, serializability, transaction, two-level serializability

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next